CS 678 Machine Learning
Project Assignment #2

Document Classification using the Naïve Bayes Algorithm

Specification

The basic idea is to write a program that, given a collection of training data consisting of forum-labeled documents, "learns" how to classify new documents using a Naïve Bayes classifier.

Background

The Naïve Bayes algorithm uses probabilities to perform classification. The probabilities are estimated based on training data for which the value of the classification is known (i.e. it is another form of Supervised Learning). The algorithm is called "naïve" because it makes the simplifying assumption that attribute values are completely independent, given the classification.

Resources

- A tutorial describing the operation of the Naïve Bayes classification algorithm has been posted on the course web page (see Naïve Bayes Classification).

Data Sets

Sample datasets have been posted on the course Web page. They are composed of documents obtained from 20 forums (or newsgroups) focusing on different topics (e.g. religion, baseball, politics). Both training (11293 documents) and test (7528 documents) data sets are provided. Dataset format:

classification documentText // one document (sample) per line

The data has been pre-processed:

- All punctuation, numbers and special characters have been removed.
- All words have been converted to lowercase.

A filtered (stemmed) version of both datasets has been further processed:

- All one- and two-character words have been removed.
- All SMART stopwords have been removed (i.e. words such as articles, adjectives, pronouns that would be expected to appear in every document). Note: this is the same technique employed by modern search engines.
- Porter's Stemmer has been applied. This process removes common English word endings; so, for example, both *biking* and *biked* would become *bik*.

Implementation

Implement the Naïve Bayes algorithm to create a document classifier

Problem: determine what class (C) a new document (D) belongs to

Approach:

- Each word position in a document is an attribute
- The value each attribute takes on is the word in that position

Simplifying assumption:

- Word probability is independent of words in other positions

Learn

1. Collect all words occurring in the Sample documents
   - Vocabulary ← set of all distinct words

2. For each class c_j (document type) in C
   - $Docs_j$ ← training documents for which the classification is c_j
   - Probability estimate of a particular class:
     $P(c_j) = | Docs_j | / |\text{training documents}|$
   - $Text_j$ ← create a single document per class (concatenate all $Docs_j$)
   - n = total number of word positions in $Text_j$
   - For each word w_k in Vocabulary
     n_k = number of times w_k occurs in $Text_j$
   - Estimate of word occurrence for particular document type:
     $P(w_k | c_j) = (n_k + 1) / (n + |\text{Vocabulary}|)$

Classify

1. Return classification C_{NB} for new document D
   - Use Naïve Bayes classifier as described in class
   - Positions ← all word positions in D containing tokens in Vocabulary (where a_i denotes word found in i^{th} position)

$$C_{NB} = \max_{c_j \in C} P(c_j) \prod_{i \in Positions} P(a_i \mid c_j)$$

Requirements

You may use either set of training/test data (original or stemmed). Obviously, train your classifier using the training data, and evaluate its effectiveness using the test data. You may modify the input files in any way you choose. You may also use any language.

Submit a written report and be prepared to present your solution to the class:

- Include complete documentation of your code.
- Describe your approach, any interesting problems encountered or experiments performed, packages used, etc.
- Calculate the effectiveness of your classifier (in other words, demonstrate that it can beat random guessing).
- Include a discussion/analysis of your results.

Further Investigation (extra credit)

- Experiment with alternative data pre-processing steps.
- Find/create a new classification problem or new datasets.
- Experiment with *k*-fold cross validation.
- Investigate/characterize the source of error in your classifier.